Practice and experience in high quality 3D graph visualization in virtual reality

Jonas Müller, Jakob Harlan, Marc Stamminger and Roberto Grosso

Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany

Corresponding authors: jonas.ej.mueller@fau.de, jakob.harlan@fau.de, marc.stamminger@fau.de and roberto.grosso@fau.de

Keywords: graph drawings, virtual reality, application

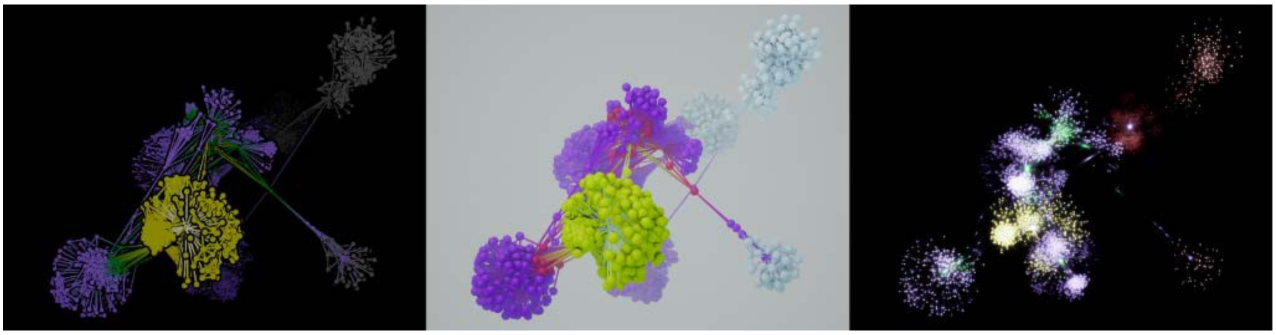

Figure 26. Screenshots of high quality 3D graph visualizations in VR: (left) flat colored drawing with halos to improve depth perception, (center) shaded solid surfaces with ambient occlusion and (right) transparent surfaces with additive blending. All configurations are applied to the same graph (Social circles: Facebook) and perspective. The color is used to visualize different graph theoretic distances to a selected node in the center of the yellow cluster in the foreground.

The visualization of complex graphs (abstract networks of nodes, connected by edges) is a long-standing research problem. Virtual reality (VR) offers a very intuitive access to the third dimension in visualization, which can help for the perception of the structure of complex graphs. However, interactive exploration of a graph in VR requires real-time rendering, which is challenging for complex graphs, even on latest high-end GPUs. Furthermore, visual effects that improve perception of the structure of the graph are desirable, but often increase render time even further.

In this paper, we describe our experiences with the high-quality rendering of 3D graphs. We show practical solutions for the efficient renderings of such graphs and for the inclusion of visual cues that improve the perception of the graph structure, without hampering real-time exploration capabilities. As a result, we present the three graph rendering techniques that are also shown in the teaser figure. We describe the real-time rendering algorithms behind these techniques, evaluate their perceptual properties, and present performance numbers on a state-of-the-art VR system.

Introduction

Graphs are widely used to represent relational data in different application areas such as software analysis, citation networks or the World Wide Web. Large graphs are visualized using node-link diagrams in different layouts or as matrix plots. Node-link diagrams are considered more instructive and popular (Ghoniem et al., 2004). Nevertheless, depending on topological structure and size of the graph, node-link diagrams can suffer from visual clutter induced by edge crossings or node overlaps. Common graph exploration tasks - such as the search for nodes with high centrality, hubs or the visual perception of high level, large scale structures such as communities - are interfered or even impeded by clutter. The problem of cluttering can be addressed by methods that locally reposition vertices or change edge shape (Wong et al., 2003; Wong and Carpendale, 2007) or globally recompute the edge routing (Holten, 2006; Holten and Wijk, 2009; Zielasko et al., 2016). Force-directed graph layouts (Fruchterman and Reingold, 1991) are often preferred because they visually represent large scale structures such as communities or clusters and are well suited for large networks. These algorithms are well understood, and efficient implementations exist. Issues with regard to computational complexity and stability of graph layouts, in particular for very large graphs were discussed in (Harel and Koren, 2001; Meyerhenke et al., 2018).

While most graph visualizations are 2D, we look into graph visualizations in 3D, in particular in virtual reality (VR). The additional dimension allows for a final layout with lower overall tension and therefore edges closer to the desired length. By displaying graphs in VR, the third, additional dimension is very intuitive to access, and the user can interact with the visualization very naturally, which can make the graph and its structure better accessible. Meanwhile, 3D graph layouts can be found in common software packages such as Gephi, e.g. Force Atlas 3D or OpenOrd 3D. Although the extension of 2D algorithms is simple and straightforward, the visual representation of graphs in 3D poses new problems like object occlusion. Thus, seen on a 2D screen, 3D layouts do not in general improve graph exploration tasks. However, Ware and Mitchell (2008) demonstrated that 3D graph layouts complemented with appropriated depth cues in an immersive virtual environment can certainly be used for graph analysis and evaluation (Halpin et al., 2008). With state-of-the-art virtual reality setups, extending this area of research is getting more reasonable, as the technology is available to a broader spectrum of users.

In this work we revisit the results proposed by Ware and Mitchell (2008) and describe our experiences when implementing a 3D graph visualization tool in VR. We demonstrate that the interactive visualization of graphs of a considerable size in an immersive virtual environment providing high quality rendering and visual cues enables users to explore and analyze their data. We propose three different real-time rendering techniques, which use different visual effects to show proximity of objects such as shadows or advanced shading to further improve intuitive understanding of the relation between data-elements.

Related Work

Graphs as given by vertices and edges edges ($G = \{V, E\}$) and their related algorithms form a large and continuously growing field in computational science. 3D graph layouts have demonstrated their applicability a time ago (Teyseyre and Campo, 2009; Brath, 2014). Commonly, data is visualized using standard 2D display where the spatial character of the visual representation is given by standard depth cues such as perspective, occlusion and shading, supported by interaction (Cutting and Vishton, 1995; Ware, 2012). As head-mounted displays become more affordable, the research on algorithms for immersive representation of non-spatial data has considerably increased in the last few years.

In a study presented by Halpin et al. (2008) the use of a 3D interface proved to significantly improve inference from non-spatial data in an immersive system. Huang et al. (2017) developed a gesture system to facilitate interaction, manipulation and analysis of graphs in a virtual environment. Erra et al. found improvements of interaction possibilities of a virtual environment compared to interactions based on mouse-keyboard or joypads when interacting with 3D graphs in their study Erra et al., 2019). Ware et al. showed that stereo vision and motion cues enable skilled users to complete the same tasks in 3D graphs up to an order of magnitude larger (Wade and Franck 1994, 1996; Ware and Mitchell, 2005). Kwon et al. (2015, 2016) have developed a spherical layout for immersive graph visualization providing a study which demonstrates the advantage of their technique. Modeling the immersive environment in a spherical space with the user's viewpoint at the center is offering a large uniform display area. Wu et al. (2006) visualize multivariate network data on the surface of a sphere by generating layouts of nodes and edges via a self-organizing map. Nevertheless, such methods even if they are on a sphere, remain 2D and are well suited for graphs of a decent size and a clear structure. Observations show that this is not the case for many important applications such as large graphs from twitter data. In order to enhance 3D perception, effects of global illumination can be added. Physically based rendering or shading often is a computational expensive task. There exist multiple methods to acquire similar effects in real-time, e.g. ambient occlusion as described by LineAO (Eichelbaum et al., 2013). Halos can help enhance perception of relative depth between lines (Everts et al., 2009, 2011; Luboschik and Schumann, 2008; van der Zwan et al., 2011). Shadowing and shading of geometry greatly assist users in understanding the geometric features. Current game-engines are a good starting point, because they already provide many of the algorithms needed for physically accurate real-time shading out-of-the-box. In this work implementations were carried out within the Unreal Engine 4.

Real-Time 3D Graph Rendering

When working with abstract graphs in general, the first step of a visualization is to generate a graph layout. The layout generation in this paper is done offline using a force-directed method (Fruchteman and Reingold, 1991) in 3D space. The layout generation is not further discussed here as it is not our focus. For 3D rendering of such graphs, most often spheres are used as the geometric representation of nodes and long-stretched cylinders as edges. Other geometric shapes such as cubes, 3D paths or glyphs can be desirable to encode further information. We will stick to spheres, but the extension to other geometry is straightforward. Render time and thus the size of a graph that can be displayed interactively in VR depend on two design choices. First, the complexity of the geometry to render the geometric objects, i.e. the spheres and cylinders. Rendering these as tessellated triangle mesh generates a huge amount of tiny geometry and is only practical for small graphs. In Section "Billboard-based rendering with Efficient Anti-Aliasing" we revisit the rendering of such primitives using billboards. Our proposed approach is significantly faster and thus allows us to render more complex graphs. Furthermore, it allows us to include anti-aliasing and thus to increase visual quality significantly in a very efficient manner. Second, we can apply more sophisticated shading techniques and special effects that improve depth and proximity perception. We consider techniques that are cheap enough to not hamper real-time rendering in Section "Shading Techniques and Special Effects". Based on these rendering techniques, we propose in Section "Proposed Render Modes" three different real-time visualization modes, which are fast enough to display complex graphs in VR, but also exhibit an improved perception of depth and proximity. In Section "Interaction in the Virtual Environment", we describe our VR interaction model for graph exploration.

Billboard-based rendering with Efficient Anti-Aliasing

Using billboards instead of complex meshes is a common strategy of reducing geometry processing complexity. To approximate spherical and cylindrical geometries, the billboard paradigm can be easily applied. For both nodes and edges, the camera-facing bounding-quadrilateral of the geometry in screen-space is calculated in the geometry pass (Figure 27). In order to obtain correct shape and shading of the geometry, accurate normals, 3D position and depth are then calculated per fragment of the billboard and outlying fragments are discarded.

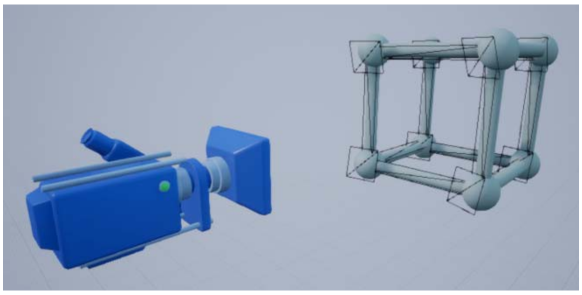

Figure 27. Billboards (black wire-frame quadrilaterals) being used instead of the actual geometry for the given camera.

For complex graphs, nodes and edges can become very small and thus will suffer from severe shader aliasing, i.e. from strong flickering. The rapid and small-scale camera-movement occurring with virtual reality magnifies the resulting flickering. For our special case of graph rendering, we propose a very simple and efficient anti-aliasing technique: Since normals are computed on-the-fly during rendering, we can adapt these normals for small structures. As soon as the footprint of the billboard in screen-space is becoming small, we smooth the normals towards the average normal. As a result, the shading becomes smoother and aliasing disappears without a visible transition, as shown in Figure 28.

When using transparent billboards, an additional transparency towards edge- or node-borders is introduced to prevent aliasing between elements. Traditional anti-aliasing methods like multi-sampling-anti-aliasing (MSAA) and temporal-anti-aliasing (TXAA) are orthogonal and can be applied additionally, e.g. to fight geometric aliasing from thin edges.

Shading Techniques and Special Effects

The shading of nodes and edges can provide important visual hints and additional depth cues. Shading with direct illumination for example helps identifying one node as unique and differentiating it from other nodes. In addition to direct shading, the relative position of nodes and edges can be clarified with advanced shading effects such as shadowing and ambient occlusion.

Ambient Occlusion

Ambient occlusion (AO) approximates indirect light transport between surfaces that are occluded by proximate geometry. In a graph environment AO serves identifying objects being close together and recognizing surface structure of larger clusters of nodes. One common and computationally cheap approach of obtaining AO is a screen-space post process (SSAO). However, with SSAO techniques, occlusion coming from objects that are not visible to the camera is missing. Furthermore, due to the high frequency changes in depth and normals typical for graph renderings, getting a stable SSAO is problematic.

With deep-screen-space, Nalbach et al. (2014) proposed a more accurate method, including occlusion from non-visible objects. Like their approach, we use splatting to accumulate the effects of ambient occlusion generated by the geometry. In addition, we directly use the properties of our sphere- and cylinder- geometry to compute analytical AO for a sphere to a surface point with the formula (Quilez, 2019):

$$I_{AO} = \max\left(0, \boldsymbol{n} \cdot \frac{\boldsymbol{d}}{|\boldsymbol{d}|}\right)\left(\frac{r}{|\boldsymbol{d}|}\right)^2$$

Where, $\boldsymbol{n}$ is the surface normal, $\boldsymbol{d}$ is the vector from the surface position to the center of the node or to the closest point on the edge and r is the radius of the node or edge. Note that, this formula is not accurate for edges, but in our experience, it still yields plausible results.

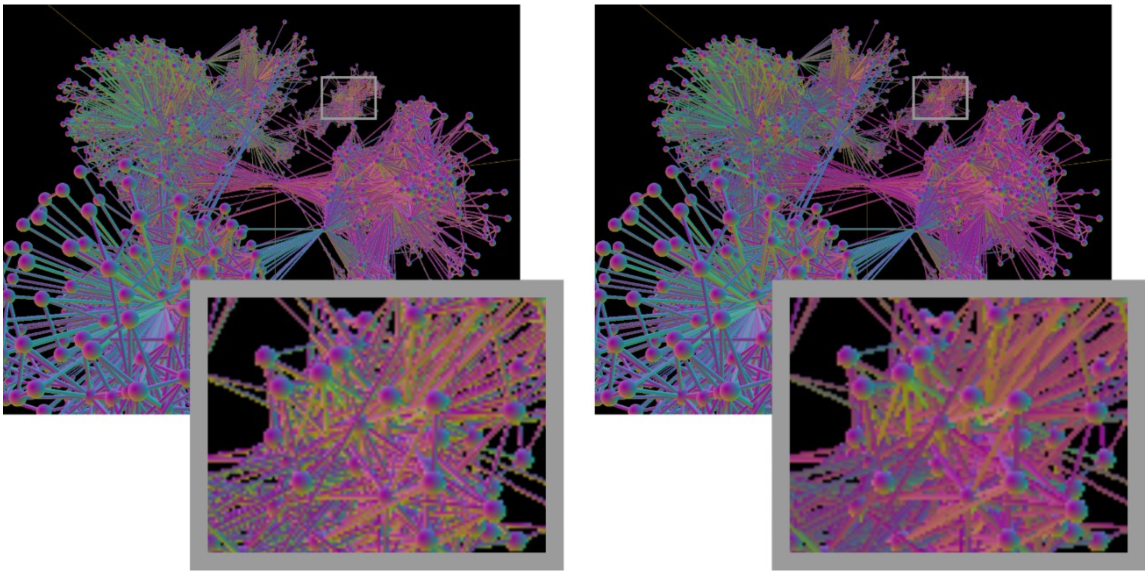

Figure 28. Elements colored by the normal at the given fragment. (left) Accurate normals showing heavy aliasing. (right) Anti-aliased normals depending on the billboard's footprint in screen-space. Note that, anti-aliasing with this method is only applied to shad

As with deep-screen-space, our splatting of ambient occlusion suffers from overestimation that is reduced by limiting the radius of influence. The splats are generated like the billboards for nodes and edges mentioned in Section "Billboard-based rendering with Efficient Anti-Aliasing". The effects of our AO-splatting (especially overestimation) and a comparison to SSAO can be seen in Figure 29.

Shadowing

Shadowing improves depth perception (Ware, 2012; Luboschik et al., 2016) (Figure 30). In our setup, shadow maps provided by the Unreal Engine are used.

Fog

The aspect of fog is receiving its own section not because of its high complexity or innovation, but because of its simplicity and yet effective application. The introduction of fog has the advantage of giving additional depth cues to the user. Furthermore, fog is a cheap way of reducing the high frequencies in the background. This eases aliasing and causes the user to focus on objects in his proximity where stereo vision works best. The application of fog can be observed in Figure 30 and Figure 32.

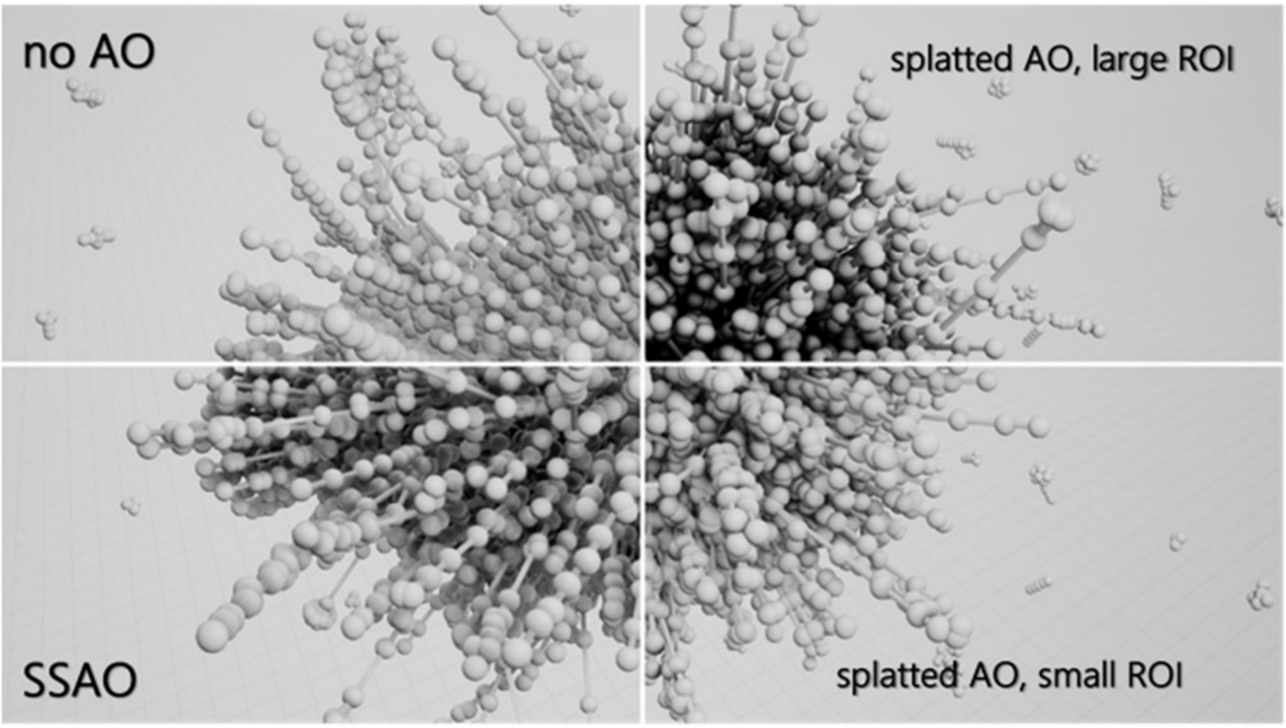

Figure 29. Ambient occlusion is giving necessary depth and proximity cues supporting the 3D perception. Direct illumination without AO (top-left). While giving a good impression, the shading of the detailed inner structure is misleading with SSAO (bottom-left). With our splatted AO (bottom-right), inter object proximity information is conveyed more accurately even for geometry not visible to the camera. When increasing the radius of influence (ROI) of our method (top-right) the problem of overestimation can be seen. Despite the lack of realism, overestimation can work as additional proximity cue - e.g. to show crowded parts like the center of the above cluster.

Halos

Depth-dependent halos Everts and Bekker, 2009) are outlines of nodes or edges, with their size varied by the depth of the scene around the object. Such halos produce a good perception of depth for 3D scenes seen on a 2D screen even for static views (Figure 32). Transparent halos (Figure 33) further improve this scenario by reducing aliasing and smoothing the otherwise sharp halo borders.

Transparency

In graph drawing, transparency can be used to overcome visual occlusion of geometry. The easiest and most widely used method of transparency in this application is realized with additive blending for example in (Royston et al., 2016). Additive edges are beneficial for localization of crowded parts of the network or dense clusters (Figure 34 and Figure 35), similar to density based approaches (Zinsmaier et al., 2012).

Proposed Render Modes

Based on the previously described rendering techniques, we derived three display modes, which are all efficient enough to display large graphs in real-time in VR and offer additional depth and proximity cues:

- Halos refers to drawing solid unilluminated geometry with halos.
- Solid & AO refers to drawing solid geometry with direct illumination, shadows and ambient occlusion.
- Additive refers to drawing additive blended transparent edges and bright nodes.

Example renderings showing these modes are shown in the teaser image, as well as in the Figure 30 to Figure 35. We will examine these three modes in more detail in the following sections "Graph perception" and "Perception" with respect to the perception of graph structures and the performance of the rendering techniques.

Interaction in the Virtual Environment

To position and orient the graph we have implemented a VR interaction suite using tracked hand-held controllers. An explanatory situation can be seen in Figure 36. Static geometry around the user (e.g. the floor) is beneficial to avoid losing orientation. To rotate and scale the network, the two-handed handlebar interaction scheme is employed (Bettio et al., 2007). The graph can be translated, rotated and scaled as if the user would grab an elastic object. Using a ray, the user can select a node to focus on. This node and its neighborhood are emphasized with different colors according to the graph theoretic distance to the selected node. In addition, the transparency of such elements is increased with increasing graph theoretic distance. This coloring scheme can be seen for example in the teaser image (ref teaser). When an illuminated visualization is chosen, a hand-held flashlight can be simulated. Having an additional light source can help in understanding compact structures by providing lighting and shadows from a user-controlled angle.

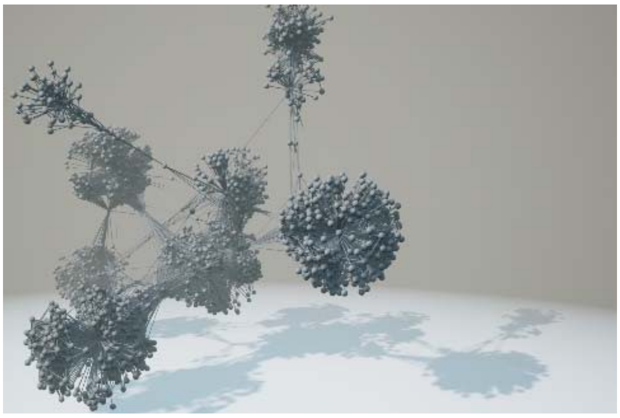

Figure 30. Shadows form an important depth cue.

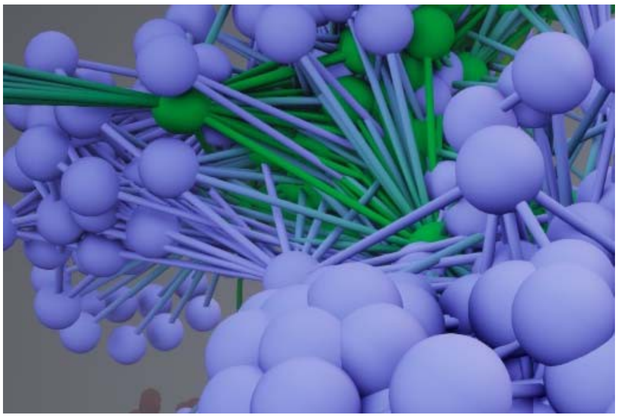

Figure 31. Depth perception for this solid & AO rendered graph is supported by ambient occlusion emitted by the edges - especially noticeable with the top green bundle of edges.

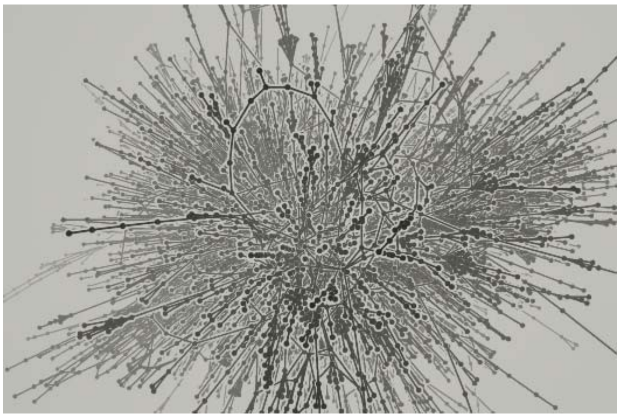

Figure 32. Illustrative halo drawing showing structure in a complex network. The relation between the nodes in the foreground can be easily observed due to the halos and the fog making them stand out from the background

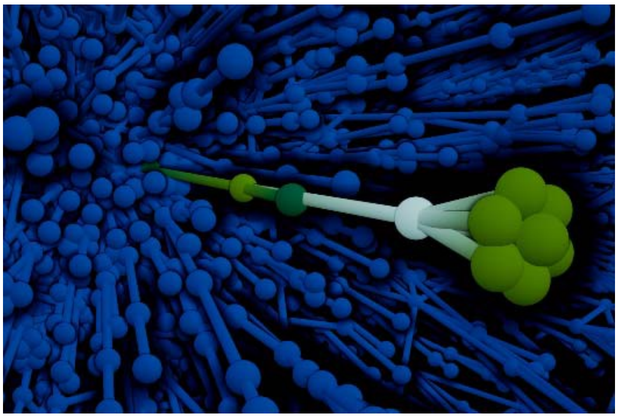

Figure 33. Transparent smooth halos deliver a good understanding of the 3D structure and depth geometry of the graph in a 2D domain like this image. In VR however, this method tends to irritate the visual system of a user.

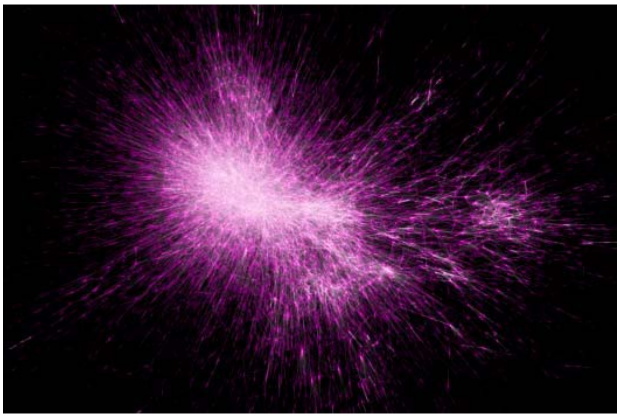

Figure 34. Edges drawn with additive blending visualize clusters and crowded areas of the graph

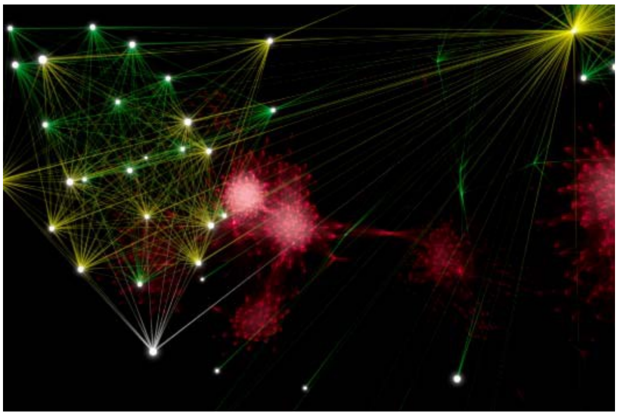

Figure 35. In additive graph drawing, close nodes are shown as bright dots, whereas in the distance only edges are drawn to remove cluttering

Figure 36. Schematic user interaction with the network structure.

Graph Perception

Evaluating the visualization's effectiveness to convey the graph information and generate excitement is a difficult task. The perception of complex structures not only depends on the visuals but also on the possible interactions a user can execute. In the following paragraphs, we present frequent feedback we got from persons who tried the system in a pre-pilot study with a size of about 10 persons and the task of freely exploring the graph at hand. As a second approach, we summarize the ease of perception based on the visible effects of each method resulting in pro and contra arguments listed in Table 1. We focus on inherent properties of the techniques that we observed looking at the participants interacting with the system and interacting with the system ourselves. The six images in Figure 30 to Figure 35 can serve as hints to retrace the arguments listed in Table 5.

Table 5. Properties observed with the collected methods that can be retraced in the figures referenced in the last column. Note that, the table reflects the authors' experience.

	Depth cues	Structural cues	Suited for stereo vision	Performance	See figure
halos	Good	No influence	Not Intuitive	Realtime	1 left, 7, 8
fog	Good	Hiding background	Good	As good as no impact	1 center, 5, 7
SSAO	Good	Missing occluded geometry	Good	Realtime	4 bottom left
splatted AO	Good	Good	Good	Interactive	1 center, 4 right, 6
solid	With shadows	With shadows	Good	Realtime	1 center, 4, 5, 6
additive	No	Clusters are emphasized	Ok	Realtime	1 right, 9, 10

Visuals

The majority of people who see the visualizations for the first time are enthusiastic about the high-quality visuals, immediately sparking their interest to take a closer look. The aesthetically pleasing presentation not only boosts engagement for the first encounter but also keeps the motivation to work with the system up, as low fidelity visuals can be a reason for frustration over time. When users switch the rendering style to the most fitting for the task at hand, their attentiveness is refreshed by the change.

Interaction

In contrast to the visuals, the users were not immediately comfortable with the two-handed interaction scheme. First, the possible operations have to be understood, then the corresponding buttons and triggers must be learned. Once a user has done this, the interactions are done fluidly. They confidently position themselves and the graph any way they want. Also selecting a node of interest and scaling graph space locally is executed without trouble.

Performance

To evaluate the achieved performance, we created simple test networks where the nodes are arranged on a regular three-dimensional grid. Adjacent nodes are connected, creating about three times more edges than nodes (Figure 37). All measurements were done on a computer equipped with an Nvidia GTX 1080 Ti graphics card, an Intel Core i7-6700 processor and 32 gigabytes of memory, rendering to an Oculus Rift. For an optimal VR experience, the frame times should be below 11 milliseconds resulting in 90 frames per second. Table 2 presents the achieved performance for rendering a network with more than 200000 nodes. Without splatted ambient occlusion we can render scenes within the specified time limit. Parameters such as the node respectively edge size have a dramatic impact on the performance as they determine the number of fragments to be processed. For the measurements reasonable settings creating a valuable visualization were chosen.

Table 6. Render times for different scenarios using the graph and perspective seen in Figure 37.

scenario	frame-time (ms)
halos	10,0
solid	9,5
solid & AO	21,7
additive	8,6

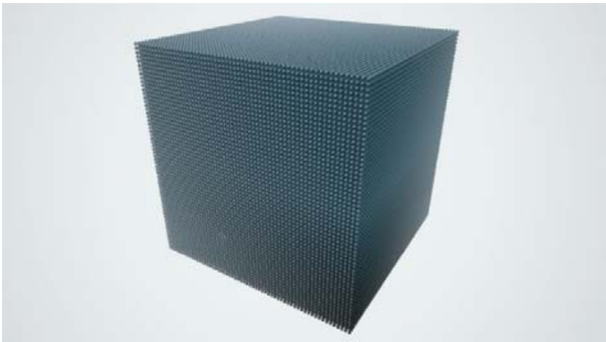

Figure 37. Synthetic grid network with 216000 nodes and 637200 edges, used to measure the timings in Table 6.

Discussion

In this work we presented three scenarios to visually represent graphs in an immersive virtual reality environment. We summarize our experience concerning the three visualization scenarios implemented as follows:

- rendering network data with halos is better suited to enhance depth cues on planar 2D screens (Figure 32). However, halos are not intuitive for human stereo vision because they have no physical equivalent. Therefore, we discourage applying them to VR.
- the rendering scenario corresponding to additive, that implements transparency, allows users to visualize large graphs due to its stable real-time performance. Furthermore, this rendering technique enables users to discern global aspects of the graph structure easily and fast (Figure 34 and Figure 35).
- the third scenario, solid & AO, implementing for example shadowing and ambient occlusion,offers the best depth cues and is therefore well suited to analyze local structure or proximity (Figure 29, Figure 30 and Figure 31). Despite being slow for larger graphs, this scenario is the most suitable for graphs in combination with the VR setup.

We showed that interactive rates for considerable large graphs can be achieved for several different visualization methods in a VR environment, which is an essential aspect for graph exploration and analysis. We put considerable effort in reducing aliasing effects, which are otherwise very disturbing especially in virtual reality. Furthermore, different intuitive interaction techniques were realized to facilitate users to explore the network data.

It would be interesting to explore how edge or node clustering can be used to combine geometry and therefore reduce the currently limiting fragment-shading bottleneck, while still providing high quality shading and enough cues to the depth and geometry perception. Furthermore, improving the performance of interactive semi-global illumination methods like the ambient occlusion presented in this work still is an important and promising research area.

References

"Social circles: Facebook," [Online]. Available: https://snap.stanford.edu/data/ego-Facebook.html. [Accessed 12 09 2019].

Bettio, F., Giachetti, A., Gobbetti, E., Marton, F. and Pintore, G., 2007. February. A Practical Vision Based Approach to Unencumbered Direct Spatial Manipulation in Virtual Worlds. In Eurographics Italian Chapter Conference, 145-150.

Brath, R., 2014, 3D InfoVis is here to stay: Deal with it. In 2014 IEEE VIS International Workshop on 3DVis (3DVis), 25-31.

Cutting, J.E. and Vishton, P.M., 1995. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In Perception of space and motion, pp. 69-117. Academic Press.

Eichelbaum, S., Hlawitschka, M. and Scheuermann, G., 2012. LineAO-Improved three-dimensional line rendering. IEEE Transactions on Visualization and Computer Graphics, 19(3), 433-445.

Erra, U., Malandrino, D. and Pepe, L., 2019. Virtual reality interfaces for interacting with three-dimensional graphs. International Journal of Human–Computer Interaction, 35(1), 75-88.

Everts, M., Bekker, H., Roerdink, J.B. and Isenberg, T., 2011, September. Illustrative line styles for flow visualization. Short Paper Proceedings of the 19th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2011, September 21--23, 2011, Kaohsiung, Taiwan), 2011.

Everts, M.H., Bekker, H., Roerdink, J.B. and Isenberg, T., 2009. Depth-dependent halos: Illustrative rendering of dense line data. IEEE Transactions on Visualization and Computer Graphics, 15(6), 1299-1306.

Fruchterman, T.M. and Reingold, E.M., 1991. Graph drawing by force-directed placement. Software: Practice and experience, 21(11), 1129-1164.

Ghoniem, M., Fekete, J.D. and Castagliola, P., 2004, October. A comparison of the readability of graphs using node-link and matrix-based representations. In IEEE Symposium on Information Visualization, 17-24.

Halpin, H., Zielinski, D.J., Brady, R. and Kelly, G., 2008, October. Exploring semantic social networks using virtual reality. In International Semantic Web Conference, 599-614. Springer, Berlin, Heidelberg.

Harel, D., and Koren, Y., 2001. A Fast Multi-scale Method for Drawing Large Graphs. In Graph Drawing, Berlin.

Holten, D. and Van Wijk, J.J., 2009. June. Force‐directed edge bundling for graph visualization. In Computer graphics forum, 28(3), 983-990. Oxford, UK: Blackwell Publishing Ltd.

Holten, D., 2006. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. IEEE Transactions on visualization and computer graphics, 12(5), 741-748.

Huang, Y.J., Fujiwara, T., Lin, Y.X., Lin, W.C. and Ma, K.L., 2017, April. A gesture system for graph visualization in virtual reality environments. In 2017 IEEE Pacific Visualization Symposium (PacificVis, 41-45.

Kwon, O.H., Muelder, C., Lee, K. and Ma, K.L., 2015., Spherical layout and rendering methods for immersive graph visualization. In 2015 IEEE Pacific Visualization Symposium (PacificVis, 63-67).

Kwon, O.H., Muelder, C., Lee, K. and Ma, K.L., 2016. A study of layout, rendering, and interaction methods for immersive graph visualization. IEEE transactions on visualization and computer graphics, 22(7), 1802-1815.

Luboschik, M. and Schumann, H., 2008, Illustrative halos in information visualization. In Proceedings of the working conference on Advanced visual interfaces, 384-387. ACM.

Luboschik, M., Berger, P. and Staadt, O., 2016, November. On spatial perception issues in augmented reality based immersive analytics. In Proceedings of the 2016 ACM Companion on Interactive Surfaces and Spaces, 47-53.

Meyerhenke, H., Nöllenburg, M. and Schulz, C., 2017. Drawing large graphs by multilevel maxent-stress optimization. IEEE transactions on visualization and computer graphics, 24(5), 1814-1827.

Nalbach, O., Ritschel, T. and Seidel, H.P., 2014, March. Deep screen space. In Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, 79-86.

Quilez, I. [Online]. Available: http://www.iquilezles.org/www/articles/spherеao/sphereao.htm. [Accessed 12 09 2019].

Royston, S., DeFanti, C. and Perlin, K., 2016. A collaborative untethered virtual reality environment for interactive social network visualization. arXiv preprint arXiv:1604.08239.

Teyseyre, A.R. and Campo, M.R., 2008. An overview of 3D software visualization. IEEE transactions on visualization and computer graphics, 15(1), 87-105.

Van Der Zwan, M., Lueks, W., Bekker, H. and Isenberg, T., 2011, June. Illustrative molecular visualization with continuous abstraction. In Computer Graphics Forum, 30(3), 683-690. Oxford, UK: Blackwell Publishing Ltd.

Ware, C. and Franck, G., 1994, October. Viewing a graph in a virtual reality display is three times as good as a 2D diagram. In Proceedings of 1994 IEEE Symposium on Visual Languages, 182-183.

Ware, C. and Franck, G., 1996. Evaluating stereo and motion cues for visualizing information nets in three dimensions. ACM Transactions on Graphics (TOG), 15(2), 121-140.

Ware, C. and Mitchell, P., 2005, Reevaluating stereo and motion cues for visualizing graphs in three dimensions. In Proceedings of the 2nd symposium on Applied perception in graphics and visualization, 51-58.

Ware, C. and Mitchell, P., 2008. Visualizing graphs in three dimensions. ACM Transactions on Applied Perception (TAP), 5(1), 2.

Ware, C., 2012. Information visualization: perception for design. Elsevier.

Wong, N. and Carpendale, S., 2007. Supporting interactive graph exploration using edge plucking. In Visualization and Data Analysis. 6495, 649508. International Society for Optics and Photonics.

Wong, N., Carpendale, S. and Greenberg, S., 2003, Edgelens: An interactive method for managing edge congestion in graphs. In IEEE Symposium on Information Visualization 2003. (IEEE Cat. No. 03TH8714, 51-58.

Wu, Y. and Takatsuka, M., 2006, Visualizing multivariate network on the surface of a sphere. In Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation.60, 77-83). Australian Computer Society, Inc.

Zielasko, D., Weyers, B., Hentschel, B. and Kuhlen, T.W., 2016, June. Interactive 3D force-directed edge bundling. In Computer Graphics Forum 35(3), 51-60).

Zinsmaier, M., Brandes, U., Deussen, O. and Strobelt, H., 2012. Interactive level-of-detail rendering of large graphs. IEEE Transactions on Visualization and Computer Graphics, 18(12), 2486-2495.